



A comparative study on forecasting polyester chips prices for 15 days, using different hybrid intelligent systems

Mojtaba Sedigh Fazli, Jean-Fabrice Lebraty

► To cite this version:

Mojtaba Sedigh Fazli, Jean-Fabrice Lebraty. A comparative study on forecasting polyester chips prices for 15 days, using different hybrid intelligent systems. International Joint Conference on Neural Networks, Aug 2013, Dallas, Texas, United States. pp.1869-1875. hal-00859445

HAL Id: hal-00859445

<https://univ-lyon3.hal.science/hal-00859445>

Submitted on 8 Sep 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A COMPARATIVE STUDY ON FORECASTING POLYESTER CHIPS PRICES FOR 15 DAYS, USING DIFFERENT HYBRID INTELLIGENT SYSTEMS

Mojtaba Sedigh Fazli¹, Jean-Fabrice LEBRATY²

ABSTRACT

Forecasting in a risky situation is a very important function for managers to assist in decision making. One of the fluctuated markets in stock exchange market is chemical market. In this research the target item for prediction is PET (Poly Ethylene Terephthalate) which is the raw material for textile industries and its very sensitive on oil prices and the demand and supply ratio. The main idea is coming through NORN model which was presented by T. Lee and James N.K. Liu in 2001. In this article after modifying the NORN model, a model has been proposed and real data are applied to this new model (we named it AHIS which stands for Adaptive Hybrid Intelligent System). Finally three different types of simulation have been conducted and compared together, which show that hybrid model which is supporting both Fuzzy Systems and Neural Networks concepts, satisfied the research question considerably. In normal situation the model forecasts a relevant trend and can be used as a DSS for a manager.

KEYWORDS:

Efficient Market Hypothesis, Financial Forecasting, Chemicals, Artificial Intelligence, Artificial Neural Networks, Decision Support System, Locally Linear Model Tree, Hybrid Neuro Fuzzy Model.

1. INTRODUCTION

By innovation of Artificial Intelligence, Financial Forecasting such as Stock Price Predictions entered in new phase. Sadly there are a lot of financial managers and experts who do not believe in forecasting but the method of AI tools which are following and predicting the time series trends is still a hot issue in management and mathematics. We think that we can capitalize on previous work in order to provide to current decision maker in a specific field an adapted decision support system. In this paper we want to answer to the following research question “How to forecast PET chips prices for 15 days?”

1.1 Why the AI methods are appropriate for this issue?

To handle this project there is 2 major categories, one is using traditional methods, in this category there is 2 major methods which are named Fundamental Analysis and Technical Analysis and the second solution is to use the novel tools such as AI tools. Because of the nature of price trends in stock markets, which is following a chaotic process [9], the research seems to be compatible drawing on AI tools. A chaotic system is including two different parts, one is stochastic and another part is deterministic, when the market trend is not too noisy the deterministic part will be more than 50%, in this case for remain part, obviously there are a lot of parameters which are affecting the price direction and fluctuations. Because of the variety of factors which are controlling and affecting the curve, it's considered that this part is stochastic and random. Artificial Neural Networks (ANNs) are the best tools for modeling nonlinear sophisticated problems and they mimic the way humans are following in solving the sophisticated problems. Since the problem solving in our brain is not a classic and algorithmic method and in human system there is a mass network of billion neurons which are working as a parallel system, each neuron is doing just one or two instruction per second, despite the sophisticated computer systems which are including a central process unit and doing a million instructions per second. For mathematicians there is a big concern about ANNs, that is they don't know how a problem will be solved (because it doesn't have an algorithm for solving the problem and just by following the brain system and adjusting the weights on synapses, it will determine the answer). But this issue is a strength point for solving a chaotic problem since in a chaotic process there is a part which has the regulation but we don't know what the regulation is.

The 70s decade was a start point for mathematicians for applying the new mathematics, time series and even some advanced tools, such as Artificial Intelligence, to verify the forecast ability of stock and other market prices. Today the prices of chemicals which are used as raw materials in lots of industries usually determine in stock exchange markets, or directly depend on some other prices which are determined in stock such as oil price, exchange rate etc. Researchers did a lot of tests and experiments on price information and stock

¹ Doctorate Student , University of Montesquieu Bordeaux 4 , Mojtabafazli@yahoo.com

² Full Professor , Lyon3 University – IAE Business school – Lab. Magellan EA3713 , jean-fabrice.lebraty@univ-lyon3.fr

exchange index in some countries such as USA, UK, Canada, Germany, Japan and etc., for finding existence or non-existence of defined structure in stock price information. At that time the most important thing for researchers was to reject the Random Walk Hypothesis [1]. Stock markets are affected and surrounded by lots of extremely interrelated parameters such as economic, social, political and even psychological indicators [2]. These mentioned indicators interacting together in a sophisticated manner, therefore it is normally very difficult and even some times impossible to forecast the fluctuations of price trends in stock markets.

There are lots of forecasting tools which are applied to this field in both the traditional and modern techniques [5]. With development of artificial intelligence researchers and investors hope that the market complexities can be untied. Previously in 90s there was a research which is conducted by Johnson and his colleagues [4] which identified a lot of potential uses of neural networks in financial institutions, corporate finance and investments. In last 20 years, the applications of the neural networks in finance solutions are increased dramatically.

2. LITERATURE REVIEW

In this section a review about the components of our research question will be presented. Firstly the notion of price behavior on a chaotic market will be explained and as a consequence a review of main AI models that are currently possible to apply for this problem will be discussed.

2.1 Efficient Market Assumption and Chaos Theory

Price behavior (especially stock price) is a challenging point which the researchers were always faced to. The main challenge is whether market price behaviors are predictable or not. Some researchers believe that prices do not follow a specific trend, rather act in a “random walk” and cannot be predicted at all [2]. They are mostly advocates of a hypothesis which is called “The Efficient Market Hypothesis (EMH)”. It has been proposed in the Efficient Market Hypothesis that in an efficient market the opportunities for profit are discovered so quickly that they seem to be opportunities [4]. Therefore there are no advantages of exclusivity and thus negating its potential performance. There has been a sense of doubt and uncertainty about the validity of the EMH, and some researchers attempted to use neural networks and other intelligent tools to validate their claims [1].

Markets are in general chaotic and usually the market curve is following chaos attitudes. A modern approach to modeling nonlinear dynamic systems like the market price trend which is fully relevant is named “*Chaos Theory*”. Chaos theory considers a process under the assumption that “part of the process is deterministic and

another part of the process is stochastic” [9]. Chaos is a nonlinear process which appears to be random. Various theoretical tests have been developed to test if a system is chaotic (has chaos in its time series). The deterministic part can be characterized using regression fitting, while the random process can be characterized by statistical parameters of a distribution function. Thus, using only deterministic or statistical techniques will not fully capture the nature of a chaotic system. Because of nature of neural networks there is an ability to capture both deterministic and random features which will make it ideal and appropriate for modeling chaotic systems [9].

2.2 MLP model

MLP model is general and most famous model of neural networks, here we will focus on optimized model of MLP which is in fact a MLP model but optimized one. This model is called GFF [8] this model of MLP has been selected because of its better performance in compare to simple MLP however its architecture is same as MLP. The architecture is illustrated as follows:

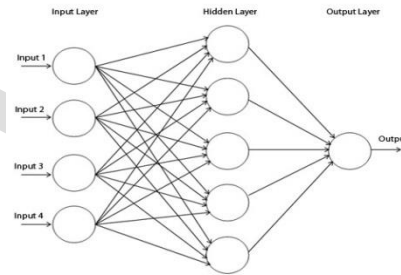


Figure 2 -3: MLP network Architecture

2.2.1 Training Process:

Training algorithm is supervised learning model. In these models input layer units distribute input signals to the network. Connection weights modify the signals that pass through it. Hidden layers and output layer are including a vector of processing elements with an activation function which is usually the sigmoid function. So that, the output of each processing unit for the forward pass will be defined as follows [8]:

$$S_i = \sum_{j=0}^n W_{ij} * u_j \quad (1)$$

$$u_i = f(S_i) \quad \text{where} \quad f(x) = \frac{1}{(1 + e^{-x})} \quad (2)$$

In the backward phase, algorithm is using the error back-propagation algorithm for weights adjustment which is using of Gradient Descent approach with a constant step length.

2.3 TDNN model

2.3.1 TDNN Model Architecture

The TDNN type which is used here for doing the simulation is called Fast Time Delay Neural Network. The model before was presented and used by Nikos Masterakis and colleagues on price prediction [9]. There is 2 different phases which must be operated respectively, learning phase and testing phase. In learning Process which is like the learning process in GFF which was explained before, but there is a different with MLP model which is applying the data to neural network through several frames instead of applying whole data in 1 step. The neurons outputs in the hidden layer multiplied by the weights of the output layer. Therefore, it may conclude that the whole problem is a cross correlation between the incoming serial data and the weights of neurons in the hidden layer [9]. Based on a mathematical theory the convolution of F with H is able to reach through following steps: suppose that F and H be the results of the Fourier Transformation of “ f ” and “ h ” in the frequency domain. Multiply F and H^* in the frequency domain point by point and then transform this product into the spatial domain via the inverse Fourier Transform. Suppose that in detection phase, a sub matrix I of size $1 \times n$ (sliding window) is extracted from the tested matrix, which has a size of $1 \times N$. Such sub matrix is fed to the neural network. Let W_i be the matrix of weights between the input sub-matrix and the hidden layer. This vector has a size of $1 \times n$ and can be represented as $1 \times n$ matrix. The output of hidden neurons $h(i)$ can be calculated as follows [9]:

$$h_i = g \left(\sum_{k=1}^n W_i(k) I(k) + b_i \right) \quad (6)$$

Where g is the activation function and b_i is dominating the bias of each hidden neuron (i). Equation 6 represents the output of each hidden neuron for a particular sub matrix I . It can be obtained to the whole input matrix Z as follows [10]:

$$h_i(u) = g \left(\sum_{k=-n/2}^{n/2} W_i(k) Z(u+k) + b_i \right) \quad (7)$$

Equation 7 represents a cross correlation operation. Given any two functions f and d , their cross correlation can be obtained by [9]:

$$d(x) \otimes f(x) = \left(\sum_{n=-\infty}^{\infty} f(x+n) d(n) \right) \quad (8)$$

Therefore, Equation 9 may be written as follows [10]:

$$h_i = g(W_i \otimes Z + b_i) \quad (9)$$

Where h_i is the output of the hidden neuron “ i ” and $h_i(u)$ is the activity of the hidden unit “ i ” when the sliding window is located at position (u) and (u) $\in [N - n + 1]$

Now, the above cross correlation can be expressed in terms of one dimensional Fast Fourier Transform as follows [10]:

$$W_i \otimes Z = F^{-1} (F(Z) \bullet F^*(W_i)) \quad (10)$$

Hence, by evaluating this cross correlation, a speed up ratio can be obtained comparable to conventional neural networks. Also, the final output of the neural network can be evaluated as follows:

$$O(u) = g \left(\sum_{i=1}^q W_o(i) h_i(u) + b_o \right) \quad (11)$$

Where, q is the number of neurons in the hidden layer. $O(u)$ is the output of the neural network when the sliding window located at the position (u) in the input matrix Z . W_o is the weight matrix between hidden and output layer.

2.4 LoLiMoT neuro fuzzy model

2.4.1 Neuro-fuzzy modeling

Here the major idea is dividing the input space into small linear subspaces with fuzzy validity functions $\phi_i(u)$ which is using for applying the hybrid locally linear neuro fuzzy model for function estimation. These functions are describing the validity of each linear model in its region [11]. The validity function applied here is the normalized Gaussian function, which is defined as:

$$\mu(x) = \exp \left(-\frac{(x-c)^2}{2\sigma^2} \right) \quad (12)$$

Where c is the center and s is the standard deviation of the Gaussian. The Gaussian function is the membership function (degree of membership of a specific object to the fuzzy sets) used in this study. Thus the total model is a neuro fuzzy network with one hidden layer and a linear neuron in the output layer which simply calculates the weighted sum of the outputs of locally linear models (LLMs) as:

$$\hat{y}_i = \omega_{i_0} + \omega_{i_1} u_1 + \omega_{i_2} u_2 + \dots + \omega_{i_p} u_p$$

$$\hat{y} = \sum_{i=1}^M \hat{y}_i \phi_i(\underline{u}) \quad (13), (14)$$

Where $\underline{u} = [u_1 \ u_2 \ \dots \ u_p]^T$ is the model input, M is the number of LLM neurons, and W_{ij} denotes the LLM parameters of the i^{th} neuron [11]. The validity functions are chosen as normalized Gaussians; normalization is necessary for a proper interpretation of validity functions:

$$\phi_i(\underline{u}) = \frac{\mu_i(\underline{u})}{\sum_{j=1}^M \mu_j(\underline{u})} \quad (17)$$

$$\mu_i(\underline{u}) = \exp \left(-\frac{1}{2} \left(\frac{(u_1 - c_{i1})^2}{\sigma_{i1}^2} + \dots + \frac{(u_p - c_{ip})^2}{\sigma_{ip}^2} \right) \right) \quad (18)$$

Each Gaussian validity function has two sets of parameters, centers (C_{ij}) and standard deviations (δ_{ij}) which are the $2M$ parameters of the nonlinear hidden layer. Optimization or learning methods are used to adjust the two sets of parameters, the rule-consequent parameters of the locally linear models (W_{ij}) and the rule premise parameters of validity functions (C_{ij} and δ_{ij}). A least squares optimization method is used to adjust the parameters of local linear models (W_{ij}), and a learning algorithm (described below) is used to adjust the parameters of validity functions (C_{ij} and δ_{ij}) [12]. Global optimization of linear parameters is simply obtained by the least squares technique. The complete parameter vector contains $M(p+1)$ elements:

$$\underline{\omega} = \begin{bmatrix} \omega_{10} & \omega_{11} & \dots & \omega_{1p} & \omega_{20} & \omega_{21} & \dots & \omega_{M0} & \dots & \omega_{Mp} \end{bmatrix}^T \quad (19)$$

And the associated regression matrix X for N measured data samples, is:

$$\underline{X} = [\underline{X}_1 \quad \underline{X}_2 \quad \dots \quad \underline{X}_M] \quad (20)$$

$$\underline{X}_i = \begin{bmatrix} \phi_i(\underline{u}(1)) & \dots & u_p(1)\phi_i(\underline{u}(1)) \\ \phi_i(\underline{u}(2)) & \dots & u_p(2)\phi_i(\underline{u}(2)) \\ \vdots & & \vdots \\ \phi_i(\underline{u}(N)) & \dots & u_p(N)\phi_i(\underline{u}(N)) \end{bmatrix} \quad (21)$$

Thus:

$$\underline{\hat{y}} = \underline{X} \cdot \underline{\hat{\omega}} \quad ; \quad \underline{\hat{\omega}} = (\underline{\hat{X}} \cdot \underline{X} + \alpha \underline{I})^{-1} \underline{X}^T \underline{y} \quad ; \quad \alpha \ll 1 \quad (22)$$

Where α is the regularization parameter for avoiding any near singularity of matrix $\underline{X}^T \underline{X}$ and in this study is empirically set to 0.001. The structure of LLNF is shown in Fig. 1. The remarkable properties of locally linear neuro fuzzy model, its transparency and intuitive construction, lead to the use of least squares technique for rule antecedent parameters and incremental learning procedures for rule consequent parameters. In this paper, Locally Linear Model Tree (LoLiMoT) algorithm as an incremental tree-based algorithm is used to tune the rule premise parameters, i.e. determining the validation hypercube for each locally linear model [12],[13]. In any iteration, the worst performing locally linear neuron is determined to be divided. All the possible divisions in the p dimensional input space are checked and the best is performed. The fuzzy validity functions for the new structure are

updated; their centers are the centers of the new hyper cubes, and the standard deviations are usually set as 0.7. For more detail refer to [13].

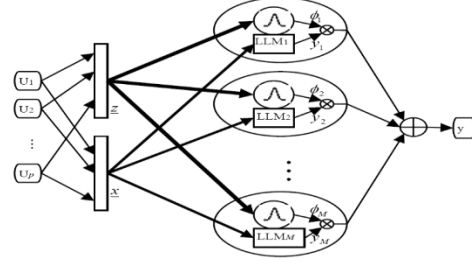


Figure 2-4: Structure of locally linear neuro-fuzzy model

2.4.2.1 Learning Algorithm

Locally Linear Model Tree (LOLIMOT) is a progressive tree construction algorithm that partitions the input space by axis bisection in all directions of input space. It implements a heuristic search for the rule premise parameters and avoids a time-consuming nonlinear optimization. The LOLIMOT algorithm is described in five steps according to [12]:

1. Start with an initial model: Start with a single LLM, which is a global linear model over the whole input space with $\phi_1(\underline{u}) = 1$, and set $M = 1$. If there is a priori input space partitioning, it can be used as the initial structure.
2. Find the worst LLM: Calculate a local loss function, for example, mean square error (MSE), for each of the $i = 1, \dots, M$, LLMs and find the worst performing LLM.
3. Check all divisions: The worst LLM is considered for further refinement. The hyper rectangle (more than a three-dimensional rectangle or cube) of this LLM is split into two halves with an axis orthogonal split. Divisions in all dimensions are tried, and for each of the p divisions, the following steps are carried out. First, construct the multidimensional membership functions for both generated hyper rectangles and construct all validity functions: In part a, only the membership function of LLM that is split would change and the membership function of other neurons do not change, but all of the validity functions change that must be updated for all LLMs by equation (17). Second, estimate the rule-consequent parameters for newly generated LLMs and third, calculate the loss function for the current overall model.
4. Find the best division: The best of the p alternatives checked in step 3 is selected, and the related validity functions and LLMs are constructed. The number of LLM neurons is incremented $M = M + 1$.

5. Test the termination condition: If the termination condition is met, then stop; otherwise, go to step 2. The termination condition is reaching to a predefined error between output (y) and LLNF output with M neuron (\hat{y}), that is, when the condition $\|y - \hat{y}\| \ll \varepsilon$ is satisfied. In practice we used a predefined number of neurons to LOLIMOT, plotted the error as a function of this number, and kept increasing the number of neurons until satisfactory performance was obtained. A suitable number of LLMs would be fit to training data on the basis of a validation set. The best number of LLMs is that in which the root mean square error (RMSE) for the validation set starts to increase. Details can be found in work by Nelles [12]. In any iteration, the worst performing locally linear neuron is determined to be divided. All the possible divisions in the p -dimensional input space are checked, and the best is selected. The splitting ratio can be simply set to 0.5, which means that the locally linear neuron is divided into two halves. The fuzzy validity functions for the new construction are updated; their centers are the centers of the new hyper cubes (more than a three-dimensional cube), and the standard deviations are usually set to 0.7 times the width of the hypercube in that dimension.

Figure 2-5 illustrates the operation of the LOLIMOT algorithm in the first four iterations for a two-dimensional input space. In iteration 1, a global linear model is fit to data. Then for refinement, input space is split into halves, and a local linear model is fit in each hyper rectangle. In iteration 2, first, the best possible splitting method is selected (e.g., in Fig.2-5, iteration 2 splitting along the u_2 axis is assumed to be better), then in the selected model, the worst LLM should be used for further refinement (shaded rectangle or 2-1, for instance), and the algorithm continues with a default number of LLMs.

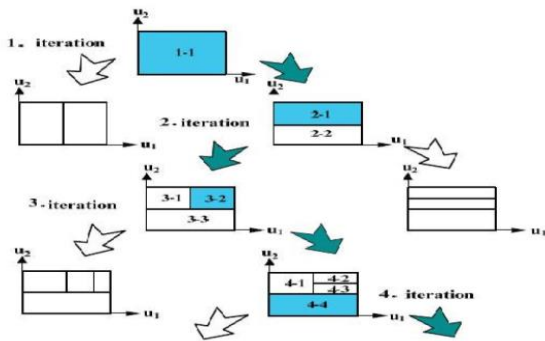


Figure 2-4: Operation of the LOLIMOT algorithm in the first five iterations for a two dimensional input space.

3. THE METHODOLOGY AND MODEL

3.1 DATA

Input data are historical data of PTA, MEG, PX and real sold price of PET Chips. They are gathered through 2 reputed sources, one is ICIS which is well known in the statistics and analysis of chemical market and another one in RECRON Company in Malaysia which is the biggest supplier of yarn in Asia, this issue is a big challenge in Asian yarn suppliers. The data set is including 347 price samples which are classified in 2 sub sets, one subset including 247 samples which are used in training process and remain 100 are used in testing process for 1 step prediction. By increasing the prediction steps to 10 and 15 days the training set size is increased and the test set is decreased.

3.2 Desired Prediction results criteria

Here there is a need to determine the acceptable error, For finding a good idea in this issue, some in depth interviews have been done with expert people in this area from east Asian chemical managers. Based on those interviews, the fitness factor and criteria could be explained as follows: If the error value which is the difference between real value and predicted value is lower than 80 USD/Ton the result is acceptable and fewer than 50 USD/Ton is desired, It means that such a different is not very crucial on this market and will not have a big effect on next item which will be produced from PET chips:

$$\text{Err} = |Fv - Rv| < 50 \quad \text{Desired}$$

$$\text{Err} = |Fv - Rv| < 80 \quad \text{Acceptable}$$

$\text{Err} = \text{Error Ratio}$, $Fv = \text{Forecasted Value}$, $Rv = \text{Real Value}$

3.3 AHIS Model

The model which is used in this research is an approach which is obtained from NORN which is presented by Ted Lee and colleagues on 2001[1]. But finally the model is different from NORN which was because of some modifications which are applied for gaining more advantages and changes also some parts of that model is eliminated and we called it AHIS which is stand for Adaptive Hybrid Intelligent System, which will make the prediction stronger and more accurate in this specific application.

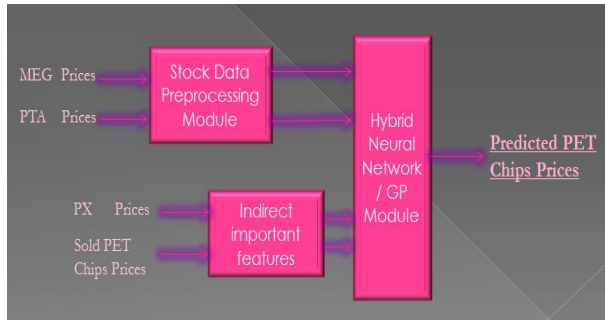


Fig 2 – 1: Research Model

3.3.1 Stock Data Preprocessing Module:

This module is doing some preprocesses which is needed to done on raw data. Here it is generally for normalizing the data.

3.3.2 Indirect important features:

In this specific application the PX price is applied as one of the important features which are indirectly affected the PET chips prices. Another feature which has been selected is sold PET chips prices, which is for considering the order and demand factor in network, since the cost price will be produce by combination of PTA and MEG in first module through considering the formula which is illustrated in Figure 2.2, here also the real market price is applied . The difference of real sold prices and cost price shows a degree of supply and demand factor.



Fig. 2 -2: Relationship between PX, PTA, MEG and PET chips

3.3.3 Hybrid Neural Network Module:

This module is the discussion point in the research, where the changes on model have been tested a lot. In Norm research, they used a recurrent NN, however in this research the MLP, TDNN, RNN, NARX, LoLiMoT models have been tested but because the NARX and RNN results was not good ,those models have been eliminated.

4. RESULTS AND DISCUSSIONS:

After designing all the models, the simulations are conducted in 3 different ways. In following simulations around 247 patterns are considered in training sets and remain 100 samples used for test set, so that the

simulation is validated for next 100 days but in just 1 step prediction. All the results are gathered in 1 picture for doing a comparison , as it obvious in figure 4 -3 , the MLP and FTDNN results were not satisfied the research question and their estimations in this specific application is not desirable :

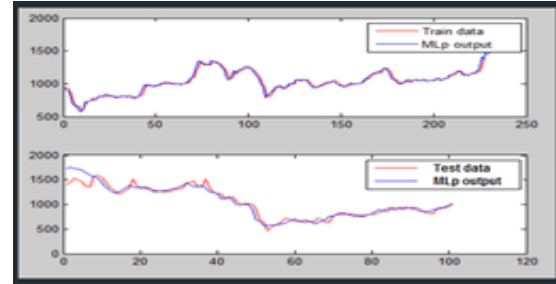


Figure 4 -1(A) : simulations results through MLP

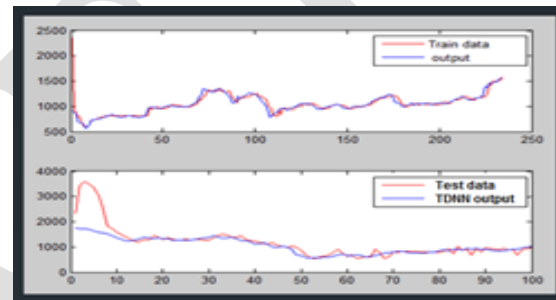


Figure 4 -1(B) : simulations results through TDNN

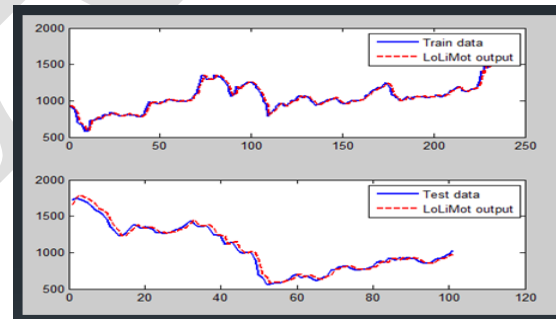


Figure 4 -2: simulations results through LoLiMoT

In above figures , the simulations which are conducted through MLP,TDNN and LoLiMoT are illustrated . each figure includes 2 parts. The top part is showing the training phase for training set and the bottom part showing the forecasted value and target value of test subset . Also for having a more tangible imagination about results ,the following table is selected to demonstrate the forecasted values and errors of each simulations for first 14 days (for simplification other 86 days have been eliminated) .

	REAL VALUE	Fv TDN N	Fv MLP	Fv LoLiM oT	TDN N Err	MLP P Err	LoLiM oT Err
Day 1	1,720.80	2,641.32	1,579.65	1,711.85	920.52	141.15	8.95
Day 2	1,757.38	2,780.25	1,562.06	1,753.40	1,022.88	195.32	3.97
Day 3	1,733.00	2,920.37	1,554.32	1,755.00	1,187.37	178.68	22.00
Day 4	1,713.00	2,921.45	1,559.43	1,710.00	1,208.45	153.57	3.00
Day 5	1,679.03	2,920.75	1,563.76	1,668.00	1,241.73	115.27	11.03
Day 6	1,642.05	2,724.14	1,571.07	1,607.00	1,082.09	70.98	35.05
Day 7	1,585.05	2,568.43	1,577.99	1,579.00	983.38	7.06	6.05
Day 8	1,565.93	1,860.97	1,580.58	1,574.00	295.05	14.65	8.08
Day 9	1,526.80	1,716.25	1,576.56	1,510.00	189.45	49.76	16.80
Day 10	1,452.88	1,572.35	1,553.53	1,401.00	119.48	100.66	51.88
Day 11	1,357.18	1,495.49	1,439.99	1,318.00	138.31	82.81	39.18
Day 12	1,298.03	1,366.97	1,387.95	1,274.00	68.94	89.93	24.03

Table 4.1) 1 step prediction values and errors of all the three simulations for first 12 days(all other 88 days have been eliminated in this table for simplification)

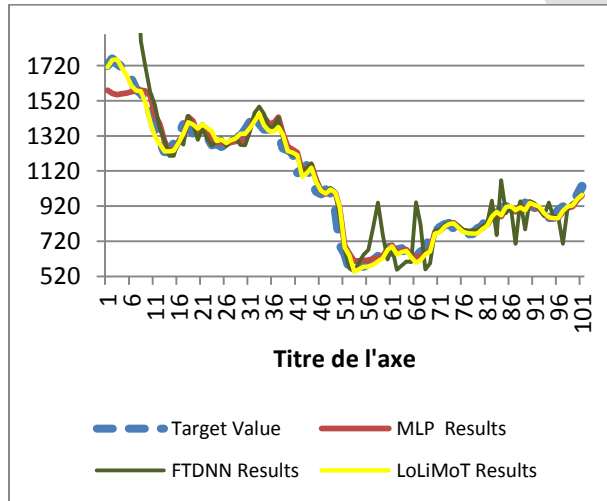


Figure 4 -3: All 3 simulations results for 100 days prediction with 1 step prediction in 1 figure

Usually for checking the error volume in such a problem, researchers are using Normalized Mean square Error which will be defined as follows:

$$NMSE = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n y_i^2} \quad (23)$$

Based on above formula the error rate of each model would be as follows:

MODEL NAME	MLP	FTDNN	LoLiMoT
NMSE RATIO	0.002831	0.07701	0.001411

Table 4 -2: NMSE ratio for around 100 day's prediction with 1 step prediction

As it mentioned in table 4-1 the error ratio for Neuro fuzzy Model is absolutely better than other 2 models, also based on desirability which was defined previously just the answers of this model are desirable. In next step just this model is tested for 5, 10 and 15 steps prediction. Also in fig. 4-4 the results for 15 steps prediction for last model which had better estimation before is illustrated , it seems that the number of patterns in this phase is not so enough ,the results for LoLiMoT is better and ultimately the results are considerable. Also in table 4-1 again the results of 15 step prediction for next 15 days are demonstrated.

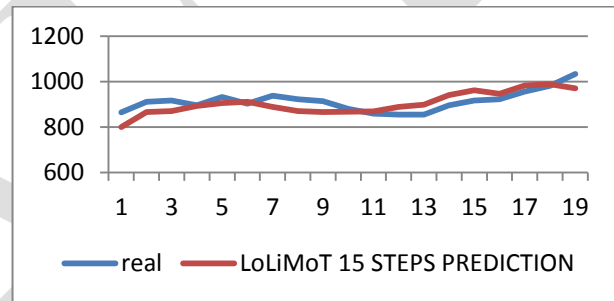


Figure 4 -4: selected simulation result for 19 days prediction with 15 step prediction

Test Day	Real Data	Predicted One	Error Volume
1	864.92	799.97	64.95
2	910.88	865.40	45.47
3	916.98	869.86	47.12
4	894.93	892.91	2.02
5	931.56	904.55	27.00
6	902.55	941.41	38.87
7	937.13	848.91	88.21
8	922.53	839.65	82.87
9	913.90	865.49	48.41
10	879.91	866.81	13.10
12	858.56	868.18	9.62
13	854.77	888.47	33.70
14	855.12	898.74	43.62
15	895.05	940.88	45.83

Table 4 -1: Forecasted value and Error volume for LoLiMoT with 15 step prediction

Finally based on formula which mentioned in (23) the NMSE ratio would be as follows:

MODEL NAME	LoLiMoT
NMSE RATIO	0.001859

Table 4 -2: NMSE ratio for around 19 days prediction with 15 step prediction

5. CONCLUSION:

In the introduction we asked “How to forecast PET chips prices for 15 Days?” We showed that AHIS including LoLiMoT (which is a hybrid neuro fuzzy model) provides a relevant answer to this question. The theoretical interest is here to propose a new model that extends the Efficient Market Hypothesis. On the managerial Interest side, this model could be embedded in a Decision Support System (DSS). Our experience in that field indicates that such tools could be very useful for real decision makers on this PET market.

This communication has some Limitations. It seems that by increasing the number of testing samples and the range of training samples and events, the system would be more stable and the answers more accurate. The last limitation is, all other models which have the potential for better answers are not yet applied such as applying Genetic programming on neural networks and using the Markov model .For further researches it's strongly offered to researchers to find a model which is combining the Markov Model with Neural Networks.

REFERENCES:

- [1] Ted Lee and James N.K.Liu, “Norn Predictor – Stock Prediction Using a Neural Oscillatory – based recurrent Network”, Imperial college Press vol. 1, No.4 (2001), p.p. 439 – 451
- [2] Abu-Mostafa YS, Atiya AF, “Introduction to financial forecasting”. Applied Intelligence 1996; 6:205–13.
- [3]Garliuskas, A. (1999). “Neural Network Chaos and computational algorithms of forecast in finance”, Proceedings of the IEEE SMC Conference, Man and Cybernetics2, PP. 638-643.
- [4] Ben krose , Patrick Van der Smag ., “ An Introduction of Neural Networks” , Amsterdam University press., Eighth edition , Nov. 1996
- [5] Zhang, D. (2005). Machine Learning Applications in Software Engineering. World Scientific Publishing.
- [6] Dase R. K, Pawar D.D, “Application of Artificial Neural Network for stock market predictions: A review of literature”, International Journal of Machine Intelligence, ISSN: 0975–2927, Volume 2, Issue 2, 2010, pp-14-17
- [7] “Basis function networks for interpolation of local linear models”, O. Nelles and R. Isermann.. IEEE Conference on Decision and Control (CDC), pages 470–475, 1996.
- [8] Rumelhart DE, Hinton GE, Williams RJ. “Learning internal representations by error Propagation. Parallel Distributed Processing: Explorations in the Microstructures of Cognition.”, McClelland, J.L. (eds.), 1: 318-362. MIT Press, Cambridge, 1986.

[9] Nikos Masterakis and colleagues, “*RECENT ADVANCES in DATA NETWORKS, COMMUNICATIONS, COMPUTERS*”, Nov. 2008, WSEAS Press, ISSN : 1790-5109, ISBN: 978-960-474-020-8

[10] Hazem M. El-Bakry, "A Novel High Speed Neural Model for Fast Pattern Recognition," Soft Computing Journal, vol. 14, no. 6, 2010, pp. 647-666.

[11] M. Nosrati Maraloo , A. R. Koushki , C. Lucas , M. M. Pedram, “Mutual Information Based Input Selection in Neuro-Fuzzy Modeling for Long Term Load Forecasting”, CSIT Proceeding , 2009.

[12] L. Ljung,” System Identification Theory for User”, Prentice-Hall, Englewood Cliffs, NJ, 1987 .

[13] O. Nelles, “*Nonlinear system identification*”, Springer Verlag, Berlin,2001.

APPENDIX I: LIST OF ABBREVIATIONS

AI	Artificial Intelligence
ANN	Artificial Neural Networks
DSS	Decision Support System
EM	Efficient Market Hypotheses
FA	Fundamental Analysis
FDY	Fully Drawn Yarn
FTDNN	Focused Time Delay Neural
GP	Genetic Programing
ICIS	Integrated chemical information system
JSE	Johannesburg Stock Exchange
LOLIMOT	Locally Linear Model Tree
MEG	Mono Ethylene Glycol
MLP	Multi-Layer Perceptron
PET	Poly Ethylene Terephthalate
POY	Partially Oriented Yarn
PTA	Purified Terephthalic Acid
PX	Paraxylene
TA	Technical Analysis